Sur la modélisation structurelle markovienne en fiabilité du logiciel

James Ledoux

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Sur la modélisation structurelle markovienne en fiabilité du logiciel

James Ledoux

N° 2714

Novembre 1995

PROGRAMME 1

ISSN 0249-6399

INRIA
RENNES

Sur la modélisation structurelle markovienne en fiabilité du logiciel

James Ledoux*

Programme 1 — Architectures parallèles, bases de données, réseaux et systèmes distribués
Projet Model

Rapport de recherche n° 2714 — Novembre 1995 — 27 pages

Résumé : La sûreté de fonctionnement est une qualité essentielle requise pour un système réparable. Nous développons ici un modèle structurel markovien dédié aux systèmes logiciels. Il permet le calcul (exact) de diverses métriques de sûreté de fonctionnement en tenant compte des éventuels temps de recouvrement de défaillances. Ces indices sont obtenus sous une forme analytique en utilisant les propriétés des processus markoviens, puis nous donnons les moyens de mettre en œuvre leur évaluation numérique. Le modèle décrit est suffisamment général pour inclure la majorité des modèles structurels markoviens connus et en particulier celui de Littlewood. Diverses questions sont également abordées à partir d'exemples simples : l'évaluation (a priori) de l'impact sur la sûreté de fonctionnement du logiciel de modifications locales d'architecture du système et d'actions de maintenance de composants (actifs ou de recouvrement). Ceci inclut, en particulier, la prise en compte de la croissance de fiabilité de certains composants. Enfin, nous donnons quelques propriétés asymptotiques du processus de défaillance.

Mots-clé : Sûreté de fonctionnement, Modèle de Littlewood, Processus MAP, Processus de comptage, Uniformisation, Croissance de fiabilité, Approximation poissonnienne

(Abstract: pto)

Une partie de ce travail a été présentée à la Rencontre Recherche-Industrie sur les modèles probabilistes et méthodes statistiques en fiabilité du logiciel. Cette journée était co-organisée par l'Association pour la Statistique et ses Utilisations (ASU) et l'IMAG de Grenoble en Juin 1995.

*. e-mail ledoux@{univ-rennes1.fr},{irisa.fr}

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 84 71 71

On the Markovian structural modeling in software reliability.

Abstract: The dependability evaluation is a basic component of the quality of a repairable system. We develop here a model which is built from a Markovian description of the behavior of a software. It allows the (exact) computation of various dependability metrics taking into account the recovery delays. These measures are given both in an analytic and algorithmic form. Questions of interest are also discussed from simple examples: how minor modifications of architecture or maintenance actions on (active or recovery) components affect the global dependability of the system. That includes in particular, to capture the reliability growth of the system from the components one. Finally, asymptotic analysis of the counting process is also addressed.

Key-words: Dependability, Littelwood's model, MAP process, Counting process, Uniformization, Reliability growth, Poisson approximation.

1 Introduction

La complexité des systèmes informatiques actuels ne permet pas d'espérer un contrôle satisfaisant de leur sûreté de fonctionnement à partir de la seule représentation boîte-noire de leur comportement. On peut constater à travers certaines études publiées que la majeure partie des modèles de type boîte-noire (pour une description complète voir par exemple [8],[25],[11],[42]) ne donne pas toujours une évaluation très précise des mesures standards de sûreté de fonctionnement. Les résultats obtenus peuvent tout de même être suffisants si on souhaite simplement appréhender une évolution de ces indices de fiabilité sans nécessairement en rechercher une estimation très précise. Il s'agit d'une situation usuelle lors des phases préliminaires de la vie d'un logiciel. La réussite des études de sûreté de fonctionnement d'un logiciel à l'aide de modèles de type boîte-noire provient généralement de la combinaison des deux étapes suivantes. On effectue un traitement préalable des données de défaillances pour identifier des plages de croissance, de décroissance et/ou de stabilité de la fiabilité [16]. Puis on sélectionne le modèle connu pour être le plus efficace pour reproduire l'ampleur de la tendance mise en évidence par les données les plus récentes. Il est important de noter que toutes ces analyses ont été réalisées post-mortem.

Par ailleurs, on souhaite fréquemment obtenir une information macroscopique sur son système à partir de connaissances microscopiques (c.a.d. au niveau du composant). En particulier, on désire connaître l'impact sur le système global d'éventuelles actions de maintenance ou de mise à jour de composants. Ce type d'attente est légitime lorsque le logiciel est en phase opérationnelle. Par exemple, on souhaite valider une modification locale de structure des opérations de recouvrement d'activité du système ou déterminer le niveau de performance nécessaire à un composant pour améliorer sensiblement la sûreté de fonctionnement globale (et par là-même sélectionner les parties du logiciel qui doivent être prioritairement revues). Le plus important, en particulier pour des problèmes de coût, est de pouvoir effectuer ces évaluations a priori, c'est à dire préalablement à toute modification effective du système. De plus, les composants des grands systèmes logiciels révèlent généralement des comportements de défaillance très différents. Par exemple, lorsqu'ils sont placés dans un environnement de type phase opérationnelle, ils représentent un ensemble d'entités soumises à des conditions de sollicitation totalement différentes de celles imposées dans les phases préliminaires d'intégration. Par conséquent, le processus de défaillance du système est fortement dépendant de la structure interne d'exécution du logiciel. Il devient alors naturel d'introduire une connaissance sur sa structure afin de mieux appréhender son comportement. La contre-partie évidente est la nécessité de collecter des données beaucoup plus riches que dans la première approche pour évaluer un tel modèle. Cette approche boîte-blanche permet de prendre en compte des dépendances entre composants de type fonctionnelles (associées à la structure du système) et de type stochastiques liées à la sûreté de fonctionnement. La modélisation par graphe d'états, en particulier par des chaînes de Markov (CM), reste la méthode la plus courante et la plus adaptée pour tenir compte de ces deux types de dépendances [35]. Elle a connu un essor considérable en raison de sa grande capacité de représentation du comportement de nombreux systèmes matériels. Le modèle markovien est un outil relativement simple pour lequel on dispose de nombreux résultats théoriques et logiciels. Il peut également être utilisé pour réaliser des analyses de coût (mesure de performabilité [6]). Enfin, il n'est pas inutile de rappeler que bon nombre de modèles de type boîte-noire (les modèle NHPP par exemple) reposent sur le caractère markovien du nombre de défaillances observées ou de fautes résiduelles (voir [11],[42]).

L'adoption d'une modélisation structurelle repose ainsi sur deux apports attendus d'un tel choix : une information collectée à partir de son modèle de meilleure qualité grâce à une plus grande représentativité du comportement du système réel et un gain sensible en quantité d'information grâce à un grain de modélisation affiné. L'objet de cette présentation est de montrer que le modèle proposé répond à ces attentes sous réserve qu'il soit adapté au système considéré. Comme nous nous intéressons ici aux modèles markoviens du comportement d'un système logiciel, nous présentons dans la prochaine section quelques conséquences d'un tel choix, que l'utilisateur pourra confronter à la réalité de son système. La Section 3 propose une étude relativement complète d'un modèle structurel markovien développé dans [24]. Ce dernier inclus, en particulier, le modèle markovien de Littlewood [27] qui reste le plus cité dans

la littérature. Nous n'abordons pas ici l'approche structurelle développée au LAAS [22],[19]. Son apport essentiel se situe dans la prise en compte de la croissance de fiabilité du système à partir de celle de ses composants. Le modèle à états est généré par un outil de modélisation standard à savoir les Réseaux de Petri Généralisés. En Section 4, nous verrons qu'il est tout à fait possible de représenter, dans notre modèle, un tel phénomène à partir de l'approche de transformation de Laprie et al. À partir de situations simples, la Section 4 apportera divers commentaires sur la sensibilité de la sûreté de fonctionnement d'un système à certaines modifications microscopiques et sur l'approximation, suggérée dans [27], du processus d'occurrence des défaillances par un modèle poissonnien.

2 Modélisation markovienne

L'évaluation de la sûreté de fonctionnement d'un système passe très souvent par l'adoption d'une représentation markovienne de son comportement ou des éventuelles caractéristiques retenues pour le calcul des indices de fiabilité. Nous allons rappeler brièvement les contraintes issues d'un tel choix. Nous supposerons ici que le système ou les grandeurs d'intérêt évoluent en temps continu, le temps étant représenté par l'ensemble $T = [0, +\infty[$; leur état à l'instant t est décrit par une variable aléatoire X_t. On s'intéresse alors au processus stochastique $X = (X_t)_{t \in T}$. Il sera caractérisé par le temps passé dans chaque état et les mécanismes de transition entre les différents états. L'évolution du processus X au cours du temps est markovienne si à toute date t, le passé du processus n'influe en rien sur son futur. Autrement dit, étant donné son état actuel X_t, le futur du processus (les valeurs de X_u pour $u > t$) est indépendant de la connaissance de X_u pour $u < t$. L'exemple fondamental qui sera développé en Section 3 (voir [27]) est celui d'un système logiciel décomposable en une famille de modules. On suppose qu'à chaque date t un seul module s'exécute. On définit alors X_t par l'identité du composant actif à l'instant t ; X_t représente donc l'état courant du processus d'exécution. Dans un tel contexte, si X_t est supposé markovien cela implique que l'identité des modules qui seront occupés à partir d'une date $u > t$ est indépendante de l'identité des modules exécutés antérieurement à t (i.e à un instant $u < t$). Un autre exemple consiste à associer X_t au nombre de défaillances recensées jusqu'à la date t. Une hypothèse markovienne revient à considérer que le nombre de défaillances que l'on aura collecté à une date $u > t$, connaissant l'état actuel du compteur, est indépendant des valeurs de ce compteur à une date antérieure à t. Maintenant, on peut établir deux classes de modèles markoviens: les chaînes de Markov homogènes (CMH) et les chaînes non-homogènes (CMNH).

2.1 Chaîne de Markov homogène

Toutes les phases d'observation d'une CMH initialisées dans un état fixé sont stochastiquement identiques (i.e deux personnes observant la CMH à partir du même état mais pas de la même date verront stochastiquement la même chose). Autrement dit, toute date t est une date de régénération (ou de renouvellement) du processus. Cette propriété d'absence de mémoire temporelle implique que si on commence à observer le processus pendant que celui-ci séjourne dans un état donné, la durée du séjour que l'on va mesurer est stochastiquement identique à celle que l'on aurait observée si la date initiale d'observation avait coïncidé avec celle de l'entrée du processus dans l'état. La distribution du temps de séjour S dans un état possède donc la propriété classique dite d'absence de mémoire

$$\mathbb{P}(S \leq t + u \mid S > t) = \mathbb{P}(S \leq u).$$

Pour toute CMH "standard" ceci implique que la distribution de S est nécessairement exponentielle et, par conséquent, que le taux de sortie de l'état est une constante qui ne dépend que de l'identité de l'état occupé. En termes de sûreté de fonctionnement une conséquence fondamentale est que si un changement d'état peut être lié à l'occurrence d'une défaillance, le taux d'occurrence de celle-ci est nécessairement

une constante indépendante du temps. Comme toute date est un point de régénération pour une CMH, nous pouvons également en tirer d'autres conclusions.

- Si on observe la chaîne seulement lors des changements d'état, chacune de ces dates étant un point de régénération, la probabilité de transiter d'un état i vers un état j est une constante qui ne dépend que des identités i et j des deux états.
- Les durées des séjours successifs dans un état donné forment une suite de variables aléatoires indépendantes, équidistribuées selon une loi exponentielle de paramètre ne dépendant que de l'état occupé, et elles sont indépendantes des temps de séjour dans un autre état.

Si nous reprenons comme exemple de modèle markovien, le processus d'exécution d'un logiciel modulaire défini par Littlewood, il possède alors les deux caractéristiques suivantes.

- La sélection du prochain module à exécuter ne dépend que du module actuellement activé.
- Les temps d'exécution d'un module ne dépendent que de l'identité du module. Ils sont indépendants des temps d'exécution associés aux modules précédemment activés et donc en particulier aux modules ayant éventuellement provoqué l'exécution du module actif.

Ainsi, l'homogénéité de la chaîne de Markov suggère une stabilité temporelle de tous les paramètres régissant les temps de séjour dans les états et les mécanismes de transition entre états (par conséquent une stabilité de tous les paramètres de défaillances). Cette classe de modèle est donc appropriée pour décrire un système en situation de fiabilité stabilisée [21] et opérant dans un environnement homogène avec le temps.

2.2 Chaîne de Markov non-homogène

Pour une CMNH, si t est la date courante d'observation, l'information nécessaire à la détermination du futur du processus dépend de l'identité de l'état actuellement occupé et du temps écoulé depuis l'initialisation du processus (à savoir de t). Il s'agit donc de modèles dont les paramètres associés aux temps de séjour et aux transitions entre états peuvent dépendre du temps écoulé depuis l'état initialement occupé par le processus (i.e X_0). Ils traduisent une non-homogénéité des paramètres régissant la dynamique du modèle. Cela paraît tout à fait attractif du point de vue de la modélisation pour représenter une non-stabilité, par exemple, des paramètres de défaillance (croissance, décroissance de fiabilité) ou une non-homogénéité de l'environnement dans lequel évolue le système. Cependant, on est très vite limité pour exploiter analytiquement et numériquement un tel modèle dans toute sa généralité. Pour quelques familles de chaînes, on peut tout de même obtenir une description assez complète des grandeurs de sûreté de fonctionnement. Si nous reprenons l'exemple du compteur de défaillances comme processus makovien, la plupart des modèles boîte-noires utilisés sont des processus de Poisson non-homogènes (NHPP) qui entrent dans la classe des CMNH [42]. Par contre, tout reste à faire pour l'exemple du processus d'exécution d'une structure modulaire. Par conséquent, dans le contexte d'une modélisation markovienne de type boîte-blanche, les processus sont quasi-systématiquement supposés homogènes. On peut tout de même introduire certaines formes de non-homogénéité tout en conservant l'essentiel de la puissance d'analyse des CMH :

- une non-homogénéité des indices de sûreté de fonctionnement avec, par exemple, la méthode de transformation [22] d'une chaîne de Markov homogène représentant le comportement du système en régime de fiabilité stabilisée ;
- Une non-homogénéité dans l'environnement pour des systèmes actifs sur des intervalles de mission [40],[34],[17].

2.3 Chaîne semi-markovienne

On peut également souhaiter conserver le caractère markovien du processus X aux seuls instants de transition entre états. Ceci permet de représenter des temps de séjour dans un état équidistribués selon une loi quelconque et ne dépendant que de l'identité de l'état occupé et du prochain état visité. On obtient ainsi une chaîne semi-markovienne. Cette classe de processus est utilisée par Littlewood dans [28] pour représenter l'évolution du processus d'exécution d'un logiciel décrit en début de section. Il en résulte une totale généralité de la distribution des temps d'exécution de chaque module au prix d'une complexité de l'évaluation analytique (et numérique) plus importante que celle présentée en Section 3 pour le cas markovien (homogène). Sachant que, théoriquement, (presque) toute distribution de probabilité sur T (représentant une durée) peut être approchée par une distribution de type phase PH (c.a.d. par la loi du temps d'absorption d'une certaine CMH [32]), on peut remplacer l'étude d'un modèle semi-markovien par celle d'une CMH. En général, le problème de la sélection des états ou phases de la CMH permettant d'approcher une distribution donnée est difficile ; il est l'objet de nombreuses études (voir par exemple [30],[7]) qui ont déjà donné des algorithmes de construction relativement stables, au moins dans le cas de deux phases. Cette idée est à la base des hypothèses retenues en Section 3 pour les temps de recouvrement de défaillance dont les distributions sont souvent peu dispersées (coefficient de variation < 1). Finalement, si le comportement du système ou des grandeurs considérées est clairement non-markovien, un procédé classique pour obtenir un modèle avec une meilleure adéquation à l'hypothèse markovienne, consiste à inclure plus d'information dans la définition de l'état du processus. Un exemple de redéfinition des états dans le contexte du modèle markovien de Littlewood est discuté dans [26]. Il s'agissait de surmonter les difficultées exposées en Sous-section 2.1, difficultées dues à l'adoption d'une hypothèse markovienne pour le processus d'exécution. L'inconvénient majeur de ce type de transformation est la génération d'un nombre important d'états ce qui augmente la complexité de la tâche d'estimation des paramètres du modèle.

3 Un modèle structurel markovien

Si les travaux de type boîte-noire sont l'objet de très nombreux articles, l'approche structurelle n'a été abordée que dans une vingtaine de publications principalement réparties entre la fin des années 70 et le début de cette décennie. Un échantillon représentatif regroupe [3],[38],[39],[14] pour une évolution discrète du temps et [27],[28],[20],[31],[18],[1],[22],[23] pour une évolution continue.

Dans un premier temps, rappelons succintement le contexte de développement du modèle markovien de Littlewood [27] qui est commun à la plupart des autres travaux cités précédemment. Ceci introduira de manière naturelle le *modèle nominal* développé dans la prochaine sous-section. L'entité élémentaire est le module. La structure du logiciel est représentée par le graphe d'appel à l'ensemble $\mathcal{M}$ de ces modules. L'interaction entre ces modules correspond à des transferts du contrôle de l'exécution et à chaque instant, ce contrôle est assuré par un et un seul module qui sera appelé le module *actif*. A partir d'une telle représentation du système, nous contruisons un processus stochastique, supposé markovien [27] (ou éventuellement semi-markovien [28]), désignant à chaque instant le module actif et modélisant la structure d'exécution du logiciel. Nous obtenons ainsi un objet mathématique possédant des propriétés bien connues. Comme nous l'avons signalé en Section 2, ceci inclut les deux caractéristiques suivantes pour la dynamique du flux de contrôle.

- La chaîne à temps discret donnant l'identité du module occupé après chaque transition est markovienne. La valeur de la probabilité de transition d'un module i vers un module j est donc indépendante de tous les modules occupés antérieurement à i.
- La durée d'exécution d'un composant est une variable aléatoire de distribution exponentielle ne dépendant que de ce composant. Elle est indépendante des temps d'exécution de tous les modules précédemment activés.

Ces conditions sont très fortes. Le décalage entre le système réel et son modèle markovien n'est pas spécifique à l'application logicielle et se retrouve, par exemple, dans presque toutes les études d'évaluation de performances des systèmes [12]. Nous proposons dans [26] une technique classique pour obtenir une meilleure adéquation à un modèle markovien: inclure une information plus riche dans la définition d'un état, ce qui correspond mathématiquement à augmenter l'espace d'état. Ce point ne sera pas développé dans ce qui suit.

Une seconde étape consiste à adjoindre les hypothèses d'oocurrence de défaillance au modèle d'exécution. Nous obtenons alors un modèle du comportement du système tel qu'il est perçu par son utilisateur. L'ensemble constitue ce que nous appelerons par la suite un *modèle opérationnel*. Dans le contexte du modèle de Littlewood, un processus des défaillances est associé à chaque type d'événement du processus d'exécution, c'est à dire une transmission de contrôle et une exécution de durée aléatoire d'un module.

- Le nombre des défaillances enregistrées durant un séjour dans le module i, est donné par un processus de Poisson de taux μ_i.
- La quantité positive $\lambda(i,j)$, pour $i,j \in \mathcal{M}$, désigne la probabilité qu'un transfert de contrôle entre les modules i et j soit défaillant.

Finalement, Littlewood étudie le processus $(D_t)_{t\geq 0}$ comptabilisant le nombre d'incidents recensés à toute date t. Si les temps d'inter-défaillance ont tendance à être beaucoup plus importants que les délais entre les échanges de contrôle, il montre alors que D_t suit une distribution de Poisson de paramètre

$$\lambda = \sum_{i\in\mathcal{M}} \pi(i)[\mu_i + \sum_{j\in\mathcal{M}} \lambda(i,j)]$$

où π désigne la distribution stationnaire associée au processus markovien d'occupation des modules. Le même type de résultat a été démontré pour des distributions quelconques de temps de séjour et non plus exponentielles [28]. Deux remarques s'imposent.

- La dynamique du processus d'exécution n'est pas perturbée par l'occurrence d'une défaillance. Le logiciel redevient instantanément opérationnel (avec probabilité 1) même si certaines formes de recouvrement d'erreurs sont mises en œuvre pour cela. Le modèle est donc identique à celui d'un système dont l'exécution suit naturellement son cours même avec l'occurrence de défaillances. Ceci ne correspond pas à un modèle opérationnel très réaliste.
- Le modélisateur ne dispose a priori d'aucune information qui permette de juger de la qualité d'une éventuelle utilisation de l'approximation poissonienne.

La Sous-section 3.3 propose une évaluation transitoire, à la fois analytique et algorithmique, d'un modèle opérationnel markovien plus riche. On ne considère ici qu'une évolution en temps continu. On se reportera à [24] pour le traitement du modèle en temps discret. Le modèle utilisé est suffisament général pour inclure l'analyse transitoire du processus de comptage associé au modèle de Littlewood. Certaines propriétés asymptotiques seront également discutées dans la Section 4.

3.1 Un modèle nominal

La première étape dans une approche structurelle pour évaluer la sûreté de fonctionnement d'un système logiciel consiste à définir un modèle d'exécution combinant des informations ou connaissances sur la structure, les données potentiellement disponibles et les mesures à évaluer. Pour le premier point, nous supposerons qu'un système peut être vu comme un ensemble de *composants* interagissant. Chaque composant est lui-même un système, etc. Cette construction récursive est stoppée lorsque le système est considéré comme atomique, soit par nature soit parce qu'une décomposition supplémentaire est jugée

inutile. Ce concept de composant logiciel est repris de [23]. Nous supposerons que les interactions entre composants sont associées aux seuls transferts de contrôle et on imposera qu'à chaque instant, le contrôle de l'exécution est la propriété d'un unique composant. La nature des interactions est donc identique à celle du modèle de Littlewood. Nous constatons que la définition d'un composant est une tâche revenant à l'utilisateur ou au concepteur, qu'elle dépend du système analysé et de l'opportunité de collecter les données requises pour construire une telle structure d'interaction. Il est clair que la première idée est d'identifier les composants au concept standard de *modules* en génie logiciel comme dans le modèle de Littlewood (et la plupart des autres publications). Puis on utilise le graphe d'appel pour représenter les interactions entre les divers modules. Mais les hypothèses markoviennes associées au modèle d'exécution deviennent alors souvent beaucoup trop irréalistes.

Nous supposerons, dans la suite, que nous disposons d'une représentation du processus d'exécution comme un processus markovien homogène à temps continu. L'ensemble $\mathcal{M} = \{1, \ldots, M\}$ désigne l'espace d'état du processus X. Les données sont contituées du générateur infinitésimal $Q = (Q(i,j))_{i,j \in \mathcal{M}}$ et de la distribution initiale $\alpha = (\alpha_1, \ldots, \alpha_M)$. Le coefficient $Q(i,j)$ est le taux de transition du composant i vers j. Il est égal au paramètre de la distribution exponentielle de la durée d'un quelconque séjour dans l'état i, pondéré par la proportion de routage vers l'état j en l'absence de tout phénomène de défaillance. La probabilité α_i représente la proportion de cycles d'exécution de premier composant actif i. Comme les cycles sont fréquemment initialisés par l'exécution d'un nombre très réduit de composants, ces probabilités sont presque toutes nulles. L'utilisateur est en général le principal responsable de la sélection des données d'entrée présentées au logiciel. Nous appelerons α, le *profil d'utilisation* des composants du logiciel. Le processus X est supposé irréductible.

3.2 Le modèle opérationnel

La description du processus de défaillance constitue la seconde composante du modèle opérationnel. Nous distinguerons deux classes de défaillances en fonction de leurs conséquences sur la dynamique du processus d'exécution. La première est composée par les défaillances dites *primaires*. Leur occurrence provoque une interruption de l'exécution qui est redémarrée, par exemple, à partir d'un point de reprise ou qui est réinitialisée à partir d'un composant d'entrée du logiciel. Tous ces événements seront qualifiés d'*événements primaires*. Nous reviendrons plus loin sur la représentation des éventuels délais engendrés par ces arrêts. La seconde classe de dysfonctionnements du système regroupe les événements dits *secondaires*. De manière générale, ce sont des incidents dont les effets sur la dynamique du processus d'exécution sont négligeables ou négligés. Il s'agit, par exemple, d'une défaillance dont les conséquences sont suffisamment mineures pour supposer que le processus d'exécution reparte instantanément d'où il se trouvait. Un autre événement secondaire est une erreur détectée non recouverte et ne générant pas un service inacceptable pour les utilisateurs. Avec le souci de ne pas alourdir la discussion, nous parlerons indistinctement d'événement ou de défaillance secondaire bien qu'un événement secondaire ne puisse pas être systématiquement associé à une défaillance du point de vue de l'utilisateur. Ces deux classes d'incidents affectent les deux modes d'évolution du processus d'exécution : exécution d'un composant et transfert de contrôle d'un composant vers un autre.

Nous disposons maintenant de données de recouvrement du processus d'exécution collectées au cours des diverses interruptions de service constatées par les utilisateurs du logiciel. Il faut que le modèle opérationnel nous permette de mesurer la qualité du service offert par le système par rapport à l'alternance continuité-suspension du processus d'exécution, i.e. la disponibilité du système. A la lumière de ce qui précède, il semble naturel d'envisager des délais consécutifs aux seules occurrences des défaillances primaires (les événements secondaires peuvent être inclus dans la discussion mais cela n'apporte aucun éclairage particulier sur l'analyse de la prochaine sous-section). D'autres mesures de la disponibilité d'un système à rapprocher des présentes considérations sont données dans [5] et [36].

Nous supposons que les temps de reprise d'une exécution sont représentés par une distribution de type phase (cf. Sous-sections 2.3 et 3.3, Exemple 3) qui permet de modéliser des durées de recouvrement

dépendant du composant ayant entraîné l'interruption de l'exécution. En effet, les taux de défaillance désignent les taux de transition d'un composant vers le "premier" état (phase) de réparation. La structure de la matrice notée R, des taux de transition entre les états transitoires de cette distribution PH, fixe alors le chemin nécessaire au retour dans un état opérationnel. L'ensemble $\mathcal{R}$ regroupe ces états transitoires et constitue ce qu'on appelera l'ensemble des *états de recouvrement* d'activité du sytème. A ceci s'ajoute la puissance de représentation des distributions PH. En effet, il est bien connu (voir [33] par exemple) que l'ensemble des lois PH est dense dans l'ensemble des distributions à support inclus dans $\mathbb{R}^+$. Si cette remarque est d'un intérêt plutôt théorique, il est clair que cette famille est suffisamment large pour décrire assez simplement de nombreux comportements. En particulier, on peut très bien représenter une loi avec un coefficient de variation plus petit que 1 avec des distributions Erlang par exemple, ce qui risque d'être une situation usuelle pour des temps de relance d'exécution.

Le processus nominal est ainsi remplacé par un processus modélisant exécution et recouvrement d'exécution. La seule condition requise est que l'alternance continuité-interruption de service soit potentiellement infinie. Les points (1), (2), (3) et (4) suivants rassemblent les hypothèses mathématiques associées aux différents événements et phases de recouvrement.

1. Durant un séjour du processus d'exécution dans l'état i, une défaillance primaire se manifeste avec un taux constant λ_i. Une phase de recouvrement est alors initialisée.

   Le nombre d'incidents secondaires durant ce séjour est donné par un processus de Poisson de taux μ_i. Ces deux types d'événements surviennent indépendamment l'un de l'autre.

2. Un événement primaire ou secondaire peut survenir lors d'une transmission de contrôle entre deux composants. S'il s'agit d'une défaillance, on parle alors d'une défaillance *d'interfaçage primaire* ou *secondaire*. Les quantités $\lambda(i,j)$ (resp. $\mu(i,j)$) désignent les probabilités d'occurrence d'un événement primaire (resp. secondaire) lors d'un transfert de contrôle d'un état i vers un état j. Lors d'une occurrence simultanée des deux types d'incidents (avec probabilité $\lambda(i,j)\mu(i,j)$), seul l'événement primaire est pris en compte. Les paramètres $\lambda(i,j)$ et $\mu(i,j)$ valent 0 pour une transition ne correspondant pas à un transfert de contrôle. L'introduction de paramètres de défaillance distincts pour un transfert de contrôle et une défaillance "interne" est nécessaire si les données de défaillance pour chaque composant ne sont pas collectées dans un environnement représentatif de l'interaction entre les composants.

3. Si un incident primaire survient durant l'exécution du composant i ou lors d'un transfert de contrôle à partir du composant i, $\alpha(i,j)$ désigne la probabilité pour que $j \in \mathcal{R}$ soit le premier état de recouvrement visité.

4. Soit $(X^*_t)_{t\geq 0}$ le processus désignant à tout instant t, soit le composant actif soit l'état de recouvrement atteint. Son espace d'état $\mathcal{E}$ est formé de la réunion de l'ensemble $\mathcal{M}$ des états associés à une activité d'un composant et de l'ensemble $\mathcal{R}$.

   Etant donnée une trajectoire de X^*, les processus de défaillance considérés ainsi que les processus de défaillance et de recouvrement d'erreurs sont indépendants les uns des autres. Nous supposons donnée la matrice S des taux de transition entre un état de recouvrement et un état opérationnel sachant que $(S+R)1^{\mathrm{T}} = 0$. L'alternance entre période opérationnelle et période non-opérationnelle est régie par le générateur suivant :

$$
\begin{array}{lcll}
Q^*(i,j) & = & Q(i,j)(1-\lambda(i,j)) & \text{si } i \neq j \text{ et } i,j \in \mathcal{M}, \\
Q^*(i,i) & = & Q(i,i) - \lambda_i & \text{si } i \in \mathcal{M}, \\
Q^*(i,j) & = & [\ \lambda_i + \sum_{k\neq i, k\in\mathcal{M}} Q(i,k)\lambda(i,k)\ ]\alpha(i,j) & \text{si } (i,j) \in \mathcal{M}\times\mathcal{R}, \\
Q^*(i,j) & = & R(i,j) & \text{si } i,j \in \mathcal{R}, \\
Q^*(i,j) & = & S(i,j) & \text{si } (i,j) \in \mathcal{R}\times\mathcal{M}.
\end{array}
$$

Il est supposé irréductible pour garantir la pérénité du modèle opérationnel dans le temps. Il représente le générateur du processus markovien X^*.

Il n'est pas inutile de souligner que les distributions $(S(i,j))_{j\in\mathcal{M}}$ $(i \in \mathcal{R})$ peuvent représenter une réorientation du flot de contrôle résultant de mécanismes de recouvrement d'erreur invoqués lors de l'occurrence de défaillances primaires. Par exemple, observons que la présence de composants critiques dont les anomalies de fonctionnement entraînent un redémarrage global du système peut être incluse dans ce modèle en posant $S(i,j) = \alpha_j, \forall j \in \mathcal{M}$, si i est la dernière phase de "recouvrement" associée au composant critique. La transition vers un nouveau composant s'effectue alors selon le profil d'utilisation α du logiciel et indépendamment de l'état critique.

Les seuls processus de défaillances peuvent être vus comme la superposition de ceux considérés dans [27] (ou [28]) et [20]. Le type d'incidents comptabilisés dans [27] correspond à celui que nous avons nommé secondaire. Nous levons cette restriction avec l'introduction des événements primaires affectant la dynamique du processus d'exécution. En effet, si nous ne tenons pas compte des délais, les distributions $(\alpha(i,j))_{j\in\mathcal{M}}$ représentent directement la réorientation du flot de contrôle vers les différents composants $j \in \mathcal{M}$ suite à une défaillance primaire "issue" de i.

3.3 Analyse du modèle

Nous nous proposons d'étudier les quantités suivantes :

- la distribution du nombre de défaillances observées sur un intervalle de temps donné ;
- l'espérance de cette distribution.

Nous rappelons que quelques propriétés asymptotiques des fonctions valeurs moyennes du nombre de défaillances seront décrites dans la Section 4. De même, quelques commentaires seront donnés quant à l'introduction du phénomène de croissance de fiabilité dans notre modèle.

La variable aléatoire D_t désigne le nombre cumulé de défaillances primaires ou secondaires sur l'intervalle $]0,t]$. A $t=0$, nous supposons le compteur D_0 initialisé à 0, i.e. $\mathbb{P}(D_0 = 0) = 1$. Le composant $i \in \mathcal{M}$ est le premier composant actif en début de phase d'observation avec probabilité α_i, i.e.

$$\forall i \in \mathcal{M} \quad \mathbb{P}(X_0^* = i) = \alpha_i \quad ; \quad \forall i \in \mathcal{R} \quad \mathbb{P}(X_0^* = i) = 0.$$

Le processus bi-dimensionnel $Op = (D_t, X_t^*)_{t\geq 0}$ constitue l'élément de base de notre étude. A partir des hypothèses d'indépendance entre les différents processus de défaillance, entre les processus de défaillance et de recouvrement, il est clair que Op est un processus markovien homogène d'espace d'état $E = \mathbb{N} \times \mathcal{E}$. Nous notons $D_t^{(p)}$ (resp. $D_t^{(s)}$) le nombre d'événements primaires (resp. secondaires) observés sur $]0,t]$. Par conséquent, nous avons par définition $D_t = D_t^{(p)} + D_t^{(s)}$ pour tout $t \geq 0$. Formellement, définissons pour tout $k \geq 0$ et $t \geq 0$:

$$\underset{i,j\in\mathcal{M}}{a(i,j)} = \begin{cases} \lim\limits_{dt\to 0} \dfrac{1}{dt}\mathbb{P}(\ (D_{t+dt} - D_t, X_{t+dt}^*) = (0,j) \mid (D_t, X_t^*) = (k,i)\ ) & \text{si } i \neq j, \\ 0 \text{ si } i = j, \end{cases}$$

$$\underset{i,j\in\mathcal{M}}{d^{(s)}(i,j)} = \lim_{dt\to 0} \frac{1}{dt}\mathbb{P}(\ (D_{t+dt} - D_t = D_{t+dt}^{(s)} - D_t^{(s)}, X_{t+dt}^*) = (1,j) \mid (D_t, X_t^*) = (k,i)\ )$$

$$\underset{i\in\mathcal{M}, j\in\mathcal{R}}{d^{(p)}(i,j)} = \lim_{dt\to 0} \frac{1}{dt}\mathbb{P}(\ (D_{t+dt} - D_t = D_{t+dt}^{(p)} - D_t^{(p)}, X_{t+dt}^*) = (1,j) \mid (D_t, X_t^*) = (k,i)\ ).$$

Si nous prenons en compte les différentes quantités définies dans les hypothèses (1), (2) et (3) du modèle opérationnel, ces taux de transition s'expriment sous la forme :

$$\begin{array}{rcll} a(i,j) & = & Q(i,j)(1-\lambda(i,j))(1-\mu(i,j)) & \text{si } i \neq j \text{ et } i,j \in \mathcal{M}, \\ a(i,i) & = & 0 & \text{si } i \in \mathcal{M}, \\ d^{(s)}(i,i) & = & \mu_i & \text{si } i \in \mathcal{M}, \\ d^{(s)}(i,j) & = & Q(i,j)[1-\lambda(i,j)]\mu(i,j) & \text{si } i \neq j \text{ et } i,j \in \mathcal{M}, \\ d^{(p)}(i,j) & = & [\,\lambda_i + \sum_{k \neq i, k \in \mathcal{M}} Q(i,k)\lambda(i,k)\,]\,\alpha(i,j) & \text{si } (i,j) \in \mathcal{M} \times \mathcal{R}. \end{array}$$

Nous obtenons ainsi un processus markovien à temps continu autorisant des transitions sans changement d'état. Pour tout $i \in \mathcal{M}$, nous pouvons vérifier que

$$\sum_{j \in \mathcal{M}} [a(i,j) + d^{(s)}(i,j)] + \sum_{j \in \mathcal{R}} d^{(p)}(i,j) = -Q(i,i) + \lambda_i + \mu_i$$

Nous noterons cette dernière valeur δ_i. Nous pouvons alors définir trois matrices A, $D^{(p)}$, $D^{(s)}$ par

$$A = (a(i,j))_{i,j \in \mathcal{M}} - \mathrm{diag}(\delta_i)_{i \in \mathcal{M}}, \quad D^{(p)} = (d^{(p)}(i,j))_{(i,j) \in \mathcal{M} \times \mathcal{R}}, \quad D^{(s)} = (d^{(s)}(i,j))_{i,j \in \mathcal{M}}.$$

Nous pouvons réécrire le générateur Q^* du processus markovien X^* sous la forme

$$Q^* = \begin{pmatrix} A + D^{(s)} & D^{(p)} \\ S & R \end{pmatrix}.$$

L'étude du processus opérationnel Op inclut celle d'un certain nombres de modèles connus lorsque nous éliminons les délais de recouvrement de sa construction. En effet, le processus de comptage des défaillances $(D_t)_{t \geq 0}$ est similaire à celui des arrivées d'un système à file d'attente considéré dans [29]. Lorsque les distributions de probabilité $(\alpha(i,j))_{j \in \mathcal{M}}$ sont de plus indépendantes de i et identiques pour chaque composant de $\mathcal{M}$, nous obtenons le processus ponctuel versatile développé par Neuts dans [33]. Les systèmes à attente forment le contexte commun de ces études et orientent leurs auteurs vers les seules propriétés stationnaires des processus ponctuels. Nous nous attachons ici à une analyse transitoire de notre modèle opérationnel qui permet alors de compléter le travail effectué par les précédents auteurs. Nous allons présenter maintenant une série de modèles opérationnels correspondant à diverses réductions de la famille des paramètres. Ces modèles seront repris ultérieurement pour illustrer ou affiner un certain nombre de résultats obtenus pour le modèle général. Le premier exemple est le modèle structurel de Littlewood décrit en introduction. Les autres sont des cas particuliers classiques du processus versatile de Neuts et ils représentent des situations intéressantes pour notre cadre de fiabilité du logiciel. Nous rappelons que nous prenons $R = S = 0$ pour ces modèles. Cela permet de les replacer plus facilement dans leur contexte habituel.

Exemple 1 (Le modèle structurel markovien de Littlewood) Dans son modèle, il n'existe aucun événement primaire, i.e.

$$\lambda_i = 0 \text{ et } \lambda(i,j) = 0 \quad \forall i,j \in \mathcal{M}.$$

Le processus des défaillances est donc formé des seules manifestations d'événements secondaires. Nous obtenons ainsi, pour $(i,j) \in \mathcal{M}$

$$A(i,j) = \begin{cases} Q(i,j)(1-\mu(i,j)) & \text{si } i \neq j, \\ -\sum_{j \neq i} Q(i,j) - \mu_i & \text{si } i = j, \end{cases}$$

$$D^{(s)}(i,j) = \begin{cases} Q(i,j)\mu(i,j) & \text{si } i \neq j, \\ \mu_i & \text{si } i = j. \end{cases}$$

Nous constatons que le générateur $Q^* = A + D^{(s)} + D^{(p)}$ du processus X^* décrivant le composant actif dans le modèle opérationnel, est égal au générateur Q du processus idéal d'exécution X. Les défaillances n'ont donc aucune influence sur l'évolution du processus d'exécution dans ce modèle.

∎

Exemple 2 (Processus de Poisson Modulé Markovien) Supposons que la probabilité d'occurrence d'une défaillance secondaire durant un transfert de contrôle est nulle dans le modèle de Littlewood. Dans ce cas, les paramètres $\mu(i,j)$ sont tous nuls pour $i,j \in \mathcal{M}$ dans les définitions des matrices A et $D^{(s)}$ de l'Exemple 1. Nous en déduisons que

$$A = Q - \mathrm{diag}(\mu_i) \quad \text{et} \quad D^{(s)} = \mathrm{diag}(\mu_i).$$

Nous obtenons ainsi un Processus de Poisson Modulé Markovien (ou MMPP) d'occurrence des événements. Cette famille de modèles est extensivement étudiée depuis deux décennies (voir [41] par exemple). Elle est utilisée, par exemple, dans l'analyse de problèmes de multiplexage de données dans les lignes de communications [10].

∎

Exemple 3 (Processus de renouvellement PH) Eliminons tous les paramètres d'occurrence d'un quelconque événement secondaire et d'une défaillance primaire lors d'un transfert de contrôle entre deux composants, i.e.

$$\mu_i = 0, \quad \mu(i,j) = \lambda(i,j) = 0 \quad \forall i,j \in \mathcal{M}.$$

Toute manifestation d'une erreur provoque une réinitialisation (instantanée) du processus d'exécution selon le profil d'utilisation α (c.a.d. $\alpha(i,j) = \alpha_j, \forall i,j \in \mathcal{M}$). Nous obtenons alors un processus de renouvellement PH. Cette classe de processus est étudiée dans [32]. Nous rappelons que la distribution d'une variable aléatoire positive, de fonction de répartition F, est de type phase [32], si elle représente la distribution du temps jusqu'à absorption d'un processus markovien fini avec un unique état absorbant, disons l'état 0; c'est à dire, s'il existe un vecteur de probabilité $(\alpha(0), \alpha)$ et un générateur de la forme

$$\begin{pmatrix} 0 & 0 \\ -G1^{\mathrm{T}} & G \end{pmatrix},$$

tel que pour tout $t \geq 0$,

$$F(t) = 1 - \alpha e^{Gt} 1^{\mathrm{T}}.$$

Le générateur G peut être considéré comme irréductible sans aucune perte de généralité [32]. Supposons qu'une absorption est instantanément suivie d'une transition vers un état transitoire j avec probabilité α_j (on suppose $\alpha(0) = 0$). Le processus ponctuel constitué des dates des visites successives à l'état instantané 0 est un processus de renouvellement d'inter-événements équidistribués selon la distribution de type phase F. Nous parlons alors d'un processus de renouvellement PH. Le processus markovien obtenu en éliminant l'état instantané admet comme générateur

$$Q^* = G - G1^{\mathrm{T}}\alpha.$$

Dans notre contexte, le sous-générateur G du processus markovien absorbant vaut

$$A = Q - \mathrm{diag}(\lambda_i)$$

et le processus X^* a pour générateur

$$Q^* = A - A1^{\mathrm{T}}\alpha.$$

∎

Le modèle de disponibilité proposé ici nous permet donc d'évaluer ces différents modèles d'occurrence de défaillances en prenant en compte des délais de recouvrement de ces incidents. Dans les Exemples 1 et 2, il faut pour cela considérer tous les événements comme primaires au sens de notre définition et imposer un redémarrage de l'activité dans le module où la défaillance s'est produite.

3.3.1 La distribution du nombre de défaillances observées sur $]0,t]$

La technique d'uniformisation d'un processus markovien à temps continu [9] constitue l'outil de base pour obtenir nos résultats analytiques. A partir d'un processus markovien en temps continu de générateur G et d'opérateur de transition e^{Gt}, on peut construire un autre processus markovien, stochastiquement équivalent au premier, d'opérateur de transition

$$\sum_{h=0}^{\infty} e^{-ut}\frac{(ut)^h}{h!}U^h \qquad \text{où} \quad u \geq \sup\{\ i : |G(i,i)|\ \} \text{ et } \ U = I + G/u.$$

Il s'agit du processus de Markov $(U_{\mathrm{Poiss}(t)})_{t\geq 0}$, où Poiss$(t)$ est la variable aléatoire "nombre d'événements sur $[0,t]$ d'un processus de Poisson (de paramètre u)" indépendante de la chaîne markovienne à temps discret $(U_h)_{h\geq 0}$ de matrice des probabilités de transition U. Cette chaîne est appelée la chaîne "uniformisée" au taux u associée au processus en temps continu. L'uniformisation ramène ainsi l'étude du processus initial à celle d'un modèle markovien pour lequel on séjourne dans tout état avec la même distribution exponentielle (i.e. on a uniformisé le taux de sortie d'un état). Si on applique cette construction au processus à temps continu Op avec le taux

$$u = \sup\{\ i \in \mathcal{M},\ j \in \mathcal{R}\ :\ |A(i,i)|, |R(j,j)|\ \},$$

alors la chaîne évoluant en temps discret $(D_h, X_h^*)_{h\geq 0}$, désignera la chaîne de Markov uniformisée relativement au taux u. Le processus X^* peut également être uniformisé au taux u et la chaîne uniformisée obtenue évolue selon la matrice des probabilités de transition

$$\widehat{Q^*} = \begin{pmatrix} I + (A + D^{(s)})/u & D^{(p)}/u \\ S/u & I + R/u \end{pmatrix}.$$

La chaîne $(D_h, X_h^*)_{h\geq 0}$ possède les mêmes propriétés mathématiques que le modèle discret développé dans [24].

Nous allons donner dans le théorème suivant une expression analytique de la fonction de répartition de la variable aléatoire D_t. La première expression sous forme exponentielle (i.e. sous la forme d'une probabilité d'absorption d'une chaîne de Markov) s'appuie intuitivement sur la représentation de la dynamique du processus alterné exécution-réparation donnée dans la Figure 1. Cette dernière représente la région du graphe de transition de la chaîne de Markov $(D_t, X_t^*)_{t\geq 0}$, éventuellement visitée à l'instant t. La probabilité de comptabiliser ici au plus 2 défaillances à la date t est alors la fonction de survie de la chaîne de Markov absorbante, finie, dont le graphe de transition entre les états transitoires est constitué de la séquence de macro-états d'extrémité la troisième instance de "A" dans la Figure 1.

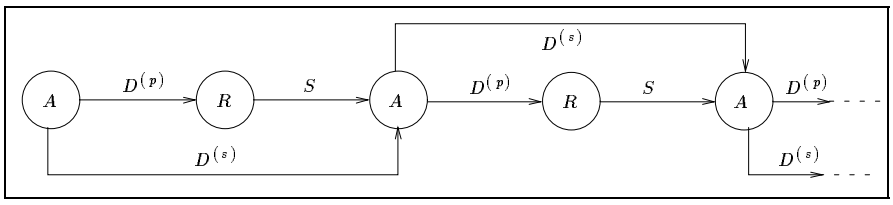


FIG. 1 - *"Graphe de transition" du processus bi-dimensionnel* (D, X^*) .

Le nombre Poiss(k,t) $(k \in \mathbb{N})$ désigne la somme

$$\mathrm{Poiss}(k,t) = \sum_{h=0}^{k} e^{-ut}\ \frac{(ut)^h}{h!}.$$

Théorème 3.1 *Pour tout $t > 0$ et $k \geq 0$, nous avons*

$$\begin{aligned} \mathbb{P}(D_t \leq k) &= (\alpha, 0)\ e^{A_{k+1}t}\ 1^{\mathrm{T}} \\ &= \mathrm{Poiss}(k,t) + \sum_{h=k+1}^{+\infty} e^{-ut}\frac{(ut)^h}{h!}\ \alpha x_{\mathcal{M}}^{\mathrm{T}}(k,h) \end{aligned} \tag{1}$$

où $u = \sup\{\ i \in \mathcal{M},\ j \in \mathcal{R}\ :\ |A(i,i)|, |R(j,j)|\ \}$ *et*

$$A_{k+1} = \begin{pmatrix} A & D^{(p)} & D^{(s)} & 0 & \cdots & \cdots & 0 \\ 0 & R & S & 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & & & \ddots & A & D^{(p)} & D^{(s)} \\ \vdots & & & & \ddots & R & S \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & A \end{pmatrix}$$

est une matrice $(2k+1) \times (2k+1)$ *par bloc définie à partir des matrices:* $A, R, S, D^{(p)}, D^{(s)}$. *Maintenant, si* $\widehat{A}, \widehat{R}, \widehat{S}, \widehat{D^{(p)}}, \widehat{D^{(s)}}$ *résultent de l'opération d'uniformisation de* $A, R, S, D^{(s)}, D^{(p)}$, *c'est à dire* $\widehat{A} = I + A/u$, $\widehat{R} = I + R/u$, $\widehat{D^{(p)}} = D^{(p)}/u$, $\widehat{D^{(s)}} = D^{(s)}/u$, $\widehat{S} = S/u$, *alors les vecteurs colonnes* $x_{\mathcal{M}}^{\mathrm{T}}(k,n,h)$ *dans l'expression* (1) *sont solutions des équations récurrentes suivantes:*

$$\left.\begin{array}{rcll} x_{\mathcal{M}}^{\mathrm{T}}(k,h) &=& \widehat{A}x_{\mathcal{M}}^{\mathrm{T}}(k,h-1) + \widehat{D^{(s)}}x_{\mathcal{M}}^{\mathrm{T}}(k-1,h-1) + \widehat{D^{(p)}}x_{\mathcal{R}}^{\mathrm{T}}(k,h-1) & h,k \geq 1, \\ x_{\mathcal{M}}^{\mathrm{T}}(0,h) &=& \widehat{A}x_{\mathcal{M}}^{\mathrm{T}}(0,h-1) & h \geq 1, \\ x_{\mathcal{M}}^{\mathrm{T}}(k,0) &=& 1^{\mathrm{T}} & k \geq 0, \\ x_{\mathcal{R}}^{\mathrm{T}}(k,h) &=& \widehat{R}\,x_{\mathcal{R}}^{\mathrm{T}}(k,h-1) + \widehat{S}x_{\mathcal{M}}^{\mathrm{T}}(k-1,h-1) & h,k \geq 1, \\ x_{\mathcal{R}}^{\mathrm{T}}(k,0) &=& 1^{\mathrm{T}} & k \geq 1. \end{array}\right\} \tag{2}$$

Le théorème précédent nous indique que le calcul de la fonction de répartition de D_t est celui de la série (1). Rappelons le calcul d'erreur inhérent à l'utilisation de la technique d'uniformisation: si nous tronquons la série (1) au seuil $H > k$ alors

$$\sum_{h=H+1}^{\infty} e^{-ut}\frac{(ut)^h}{h!}\alpha x_{\mathcal{M}}^{\mathrm{T}}(k,h) \leq \sum_{h=H+1}^{\infty} e^{-ut}\frac{(ut)^h}{h!} \quad (\text{car } \alpha x_{\mathcal{M}}^{\mathrm{T}}(k,h) \leq 1).$$

Pour un seuil d'erreur d'approximation ε de la série, on choisit donc H de tel sorte que le reste d'ordre $H+1$ de la somme de Poisson soit borné par ε. Finalement, l'algorithme se résume de la façon suivante.

(a) Choisir la borne d'erreur d'approximation ε.

(b) Calculer la valeur de H réalisant $\sum_{h=H+1}^{+\infty} e^{-ut}\frac{(ut)^h}{h!} < \varepsilon$.

(c) Calculer les vecteurs $x_{\mathcal{M}}^{\mathrm{T}}(k,h)$ pour $h = 0, \ldots, H$ avec le système de récurrences matricielles (2).

Temps jusqu'à première défaillance

Le temps jusqu'à première défaillance T dans le modèle opérationnel Op est clairement une distribution PH à partir des seconde et troisième équations de renouvellement de (2) :

$$\mathbb{P}(T > t) = \mathbb{P}(D_t = 0) = \alpha e^{At} 1^{\mathrm{T}}.$$

L'évaluation proposée par l'algorithme précédent est alors équivalente au calcul classique du vecteur de probabilité d'état d'un processus markovien absorbant au moyen de l'uniformisation.

Disponibilité instantanée

La fonction disponibilité instantanéee est définie, à la date t, par la probabilité $A(t)$ pour que le système soit opérationnel à t. Il s'agit donc de la probabilité pour que le processus identité X^* désigne un des composants (un élément de $\mathcal{M}$) à l'instant t :

$$A(t) = \sum_{i \in \mathcal{M}} \mathbb{P}(X_t^* = i) = \sum_{i \in \mathcal{M}} (\alpha, 0) e^{Q^* t}(i).$$

Comme X^* est un processus markovien irréductible, lorsque t tend vers l'infini, le vecteur ligne $(\alpha, 0)e^{Q^* t}$ converge, composante par composante, vers la distribution stationnaire π. On retrouve alors l'expression de la disponibilité (instantanée) asymptotique à partir d'un modèle markovien :

$$\lim_{t \to +\infty} A(t) = A(\infty) = \sum_{i \in \mathcal{M}} \pi(i).$$

3.3.2 Le nombre espéré de défaillances sur $]0, t]$

Définissons le vecteur colonne D^{T} sur l'ensemble $\mathcal{E}$ par

$$D^{\mathrm{T}} = \begin{pmatrix} (D^{(p)} + D^{(s)}) 1^{\mathrm{T}} \\ 0 \end{pmatrix}.$$

La valeur $D^{(p)}(i)$ $(i \in \mathcal{M})$ donne le taux d'occurrence d'une défaillance "à partir" du composant i. Le vecteur $\widehat{D^{\mathrm{T}}}$ désignera D^{T}/u.

L' espérance de la variable aléatoire D_t s'obtient en calculant directement la somme $\sum_{k \geq 0} \mathbb{P}(D_t > k)$ avec la formule (1) : pour $t > 0$,

$$\mathbb{E}[D_t] = \sum_{h=1}^{\infty} e^{-ut} \frac{(ut)^h}{h!} (\alpha, 0) \left(\sum_{k=0}^{h-1} \widehat{Q^*}^k \right) \widehat{D^{\mathrm{T}}}. \tag{3}$$

Nous avons de nouveau la somme d'une série à évaluer. L'algorithme pour calculer le nombre moyen de défaillances sur l'intervalle $]0, t]$ est constitué de trois étapes.

(a) Choisir le seuil d'erreur d'approximation ε.

(b) Calculer la valeur de H satisfaisant $ut \sum_{h=H}^{\infty} e^{-ut} \frac{(ut)^h}{h!} < \varepsilon$.

(c) Pour chaque $h = 1, \ldots, H$, calculer les sommes partielles

$$(\alpha, 0) \left(\sum_{k=0}^{h-1} \widehat{Q^*}^k \right) \widehat{D^{\mathrm{T}}}$$

pondérant les termes de Poisson dans l'expression (3).

On peut dériver des expressions de $\mathbb{E}[D_t^{(p)}]$ et $\mathbb{E}[D_t^{(s)}]$, simplement à partir des relations $\mathbb{E}[D_t^{(p)}] = \mathbb{E}[D_t] \big|_{D^{(s)}=0}$ and $\mathbb{E}[D_t^{(s)}] = \mathbb{E}[D_t] - \mathbb{E}[D_t^{(p)}]$.

Fonction intensité de défaillance

Il est facile de calculer la valeur $h(t)$ de la fonction intensité de défaillance en conditionnant par rapport à l'état occupé par le processus markovien X^* à l'instant t :

$$\begin{aligned} h(t) &= \lim_{dt \to 0} \frac{1}{dt} \mathbb{P}(D_{t+dt} - D_t = 1) \\ &= (\alpha, 0) e^{Q^* t} D^{\mathrm{T}}. \end{aligned}$$

On remarque que la fonction intensité converge avec t vers la valeur $\lambda^* = \sum_{i \in \mathcal{M}} \pi(i)[\sum_{j \in \mathcal{M}} D^{(s)}(i,j) + \sum_{j \in \mathcal{R}} D^{(p)}(i,j)]$.

L'expression de la fonction intensité nous permet de donner une autre représentation de $\mathbb{E}[D_t]$:

$$\mathbb{E}[D_t] = \int_0^t h(s) ds = (\alpha, 0) \int_0^t e^{Q^* s} ds D^{\mathrm{T}} = \sum_{i \in \mathcal{M}} \mathbb{E}[S_i(t)] \, [\sum_{j \in \mathcal{M}} D^{(s)}(i,j) + \sum_{j \in \mathcal{R}} D^{(p)}(i,j)] \qquad (4)$$

où $\mathbb{E}[S_i(t)]$ représente le temps moyen d'exécution du composant i sur $]0,t]$.

Lorsque nous négligeons les délais, nous retrouvons l'expression des fonctions densité de renouvellement et nombre moyen de renouvellements pour le processus de renouvellement PH de l'Exemple 3 (voir [32] par exemple). Nous obtenons finalement à partir de (4), pour tout $t > 0$,

$$\mathbb{E}[D_t] = (\pi D^{\mathrm{T}}) t + (\alpha, 0)(I - e^{Q^* t})(1^{\mathrm{T}} \pi - Q^*)^{-1} D^{\mathrm{T}}. \qquad (5)$$

Conformément à la terminologie adoptée par Neuts, le taux

$$\lambda_d^* = \pi D^{\mathrm{T}}$$

est appelé le *taux fondamental* (ici) d'occurrence des défaillances. Nous remarquons que pour le modèle de Littlewood (avec $R = S = 0$) il vaut

$$\lambda_d^* = \sum_{i \in \mathcal{M}} \pi(i) \, [\sum_{j \neq i} Q(i,j) \mu(i,j) + \mu_i]$$

où π désigne la distribution stationnaire du processus d'exécution. Il représente le paramètre du processus de Poisson donné comme approximation dans [27]. Pour les modèles MMPP et renouvellement PH, nous obtenons respectivement

$$\sum_{i \in \mathcal{M}} \pi(i) \mu_i \quad \text{et} \quad \sum_{i \in \mathcal{M}} \pi(i) \lambda_i.$$

Remarque 1 Dans une évaluation de sûreté de fonctionnement, il est concevable que la prise en compte d'incidents n'ayant pas de conséquence grave sur les services fournis aux utilisateurs puisse être considérée comme superflue. Si nous ne nous intéressons qu'au processus des événements primaires, il suffit alors d'annuler tous les paramètres liés à l'occurrence de défaillances secondaires (i.e $D^{(s)} = 0$). En effet, ceux-ci n'ont aucune influence sur la dynamique du modèle opérationnel et n'interviennent donc en rien dans le comportement du processus markovien X^* décrivant les états successivement visités. Si $D_t^{(p)}$ représente la variable "nombre cumulé de défaillances primaires" alors les expressions de la fonction de répartition et l'espérance associées à la variable $D_t^{(p)}$ restent identiques à celles du Théorème 3.1 et de (3) :

$$\mathbb{P}(D_t^{(p)} \leq k) = (\alpha, 0) \; e^{A_{k+1} t} \; 1^{\mathrm{T}};$$

$$\mathbb{E}[D_t^{(p)}] = \sum_{h=1}^{\infty} e^{-ut} \frac{(ut)^h}{h!} (\alpha, 0) \left(\sum_{k=0}^{h-1} \widehat{Q^*}^k \right) \widehat{D^{\mathrm{T}}},$$

où les paramètres $(\mu(i,j), \mu_i)_{i,j\in\mathcal{M}}$, ont été annulés dans les définitions des matrices $A, D^{(s)}$ pour construire le sous-générateur A_{k+1}.

Cette remarque s'applique par ailleurs au modèle de Littlewood (Exemple 3) pour obtenir l'évaluation du compteur de défaillances lié à un sous-ensemble de composants du logiciel. En effet, comme tous les incidents n'ont aucune répercution sur le processus d'exécution, i.e. $X^* = X$, il suffit de mettre à 0 tous les paramètres associés aux événements sans intérêt pour la partie du système visée. Par exemple, si on ne s'intéresse qu'aux incidents survenus dans le composant i ou qui sont issus d'une transmission de contrôle assurée par ce même composant, alors on annule tous les paramètres

$$\mu_k, \quad \mu(k,j) \quad k \neq i, j \in \mathcal{M},$$

et on utilise les formules du Théorème 3.1 et (3).

△

Remarque 2 Comme dans [24], nous pouvons distinguer dans le processus d'exécution idéal, les transitions entre deux composants i et j représentant la continuité d'un cycle d'exécution, de celles associées à la fin d'un tel cycle suivi de la transmission du contrôle au composant j. Nous adjoignons à chaque composant i, une probabilité $p_f(i)$ pour qu'il constitue le dernier composant exécuté dans un cycle d'exécution. Cette probabilité peut être interprétée comme la proportion de services délivrés une fois le composant i exécuté. Dans ce contexte, on peut calculer, par exemple, la distribution jointe du nombre de services délivrés et du nombre de défaillances observées sur $]0,t]$. Nous renvoyons à [24] pour plus de détails.

△

4 Divers commentaires

4.1 Analyse d'un modèle

Dans cette sous-section, nous illustrons brièvement les résultats du Théorème 3.1 par un exemple simple. Supposons que notre système logiciel soit constitué de 5 composants intitulés C_i, $(i = 1..5)$ dont les paramètres des temps d'exécution exponentiels ont été estimés à

$-Q(1,1)$	$-Q(2,2)$	$-Q(3,3)$	$-Q(4,4)$	$-Q(5,5)$
1	0.5	0.5	1	1

où l'unité est l'heure. La Figure 2, restreinte aux sommets associés aux 5 composants, donne le graphe de transition du processus d'exécution X du système (pour les deux modèles opérationnels présentés plus loin).

Nous avons choisi $Q(1,2) = 1$, $Q(2,3) = 0.025$, $Q(2,4) = 0.475$ et $Q(3,5) = 0.5$ pour les taux de transition entre composants.

Pour simplifier, nous ne considèrerons que l'occurrence d'événements primaires, le taux de défaillance pour chaque composant intégrant l'apparition de défaillances internes et de transfert. Les éléments C_4 et C_5 sont les seuls pour lesquels on dispose de taux de défaillance non nuls :

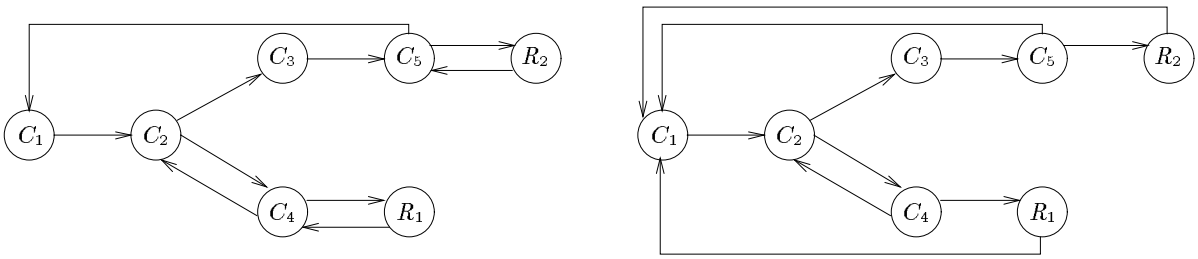
λ_4	λ_5
0.03	0.01

.

Après un incident dans C_4 (resp. C_5), les temps de recouvrement d'un mode de bon fonctionnement du système seront distribués selon une loi exponentielle de paramètre $-R(1,1)$ (resp. $-R(2,2)$) donné par :

$-R(1,1)$	$-R(2,2)$
5	10

.



Modèle $(Op)_1$ Modèle $(Op)_2$

FIG. 2 - *Le processus opérationnel du modèle à 5 composants et 2 états de recouvrement*

Les états de recouvrement seront représentés par R_1 et R_2.

Nous pouvons définir deux modèles opérationnels $(Op)_1$ et $(Op)_2$ à partir de conditions distinctes de retour à l'exécution d'un composant suite à une défaillance. Nous allons supposer pour $(Op)_1$ que l'exécution reprend à partir du composant ayant défailli. C'est l'hypothèse retenue dans [28]. Le graphe de transition de $(Op)_1$ est décrit dans la partie gauche de la Figure 2. Pour le second modèle opérationnel $(Op)_2$, l'exécution du système reprendra systématiquement dans C_1. Le graphe de transition obtenu est donné dans la partie droite de la Figure 2.

L'analyse des deux modèles est réalisée à l'aide des matrices :

$$A = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -0.5 & 0.025 & 0.475 & 0 \\ 0 & 0 & -0.5 & 0 & 0.5 \\ 0 & 1 & 0 & -1.03 & 0 \\ 1 & 0 & 0 & 0 & -1.01 \end{pmatrix} \quad D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.03 & 0 \\ 0.01 & 0 \end{pmatrix} \quad R = \begin{pmatrix} -5 & 0 \\ 0 & -10 \end{pmatrix},$$

$$S_1 = \begin{pmatrix} 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 10 \end{pmatrix} \text{ pour } (Op)_1 \text{ et } \quad S_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 10 \end{pmatrix} \text{ pour } (Op)_2.$$

Le Théorème 3.1 nous permet de calculer la fonction de répartition du nombre de défaillances sur l'intervalle $]0,t]$. La Figure 3 donne le graphe de la fonction $t \to \mathbb{P}(D_t \leq k)$ pour $(Op)_1$ (pour $k = 0, \ldots, 3$). La description complète de la loi de D_t nous permet d'évaluer un nombre très important d'indices de sûreté de fonctionnement. Pour $k = 0$, nous avons la caractéristique 0-faute du système, à savoir sa fonction fiabilité. On peut répondre également à une question telle que "jusqu'à quelle date t la probabilité d'observer au plus deux défaillances est plus grande que 0.95 ?" En utilisant l'algorithme du Théorème 3.1, on obtient $t \leq 88h$.

Le modèle structurel développé est un outil d'analyse de la sensibilité de la sûreté de fonctionnement du système global aux variations de caractéristiques de ses composants. Supposons que l'on souhaite obtenir de meilleurs indicateurs de fiabilité pour le modèle $(Op)_1$. Par exemple, on désire une probabilité d'observer au plus deux défaillances plus grande que 0.95 sur une période d'au moins dix jours (i.e. sur $]0, t_{inf}]$ avec $t_{inf} = 240h$). Seuls les composants C_4 et C_5 produisent des défaillances selon les paramètres λ_4 et λ_5. On peut alors se demander sur quel composant il est plus intéressant de travailler pour parvenir au nouvel objectif fixé de sûreté de fonctionnement et dans quelle proportion. L'exemple traité ici étant particulièrement simple, on remarque directement que C_4 est le composant pertinent. Les calculs confirment ce choix car une diminution du taux de défaillance résiduel λ_5 dans un rapport de 10 n'autorise qu'un gain de deux heures sur la borne inférieure initiale (i.e. $t_{inf} \simeq 90h$). Par contre, la simple division par trois du taux λ_4 donne une borne de $t_{inf} \simeq 257h$ qui assure l'objectif fixé. L'allure

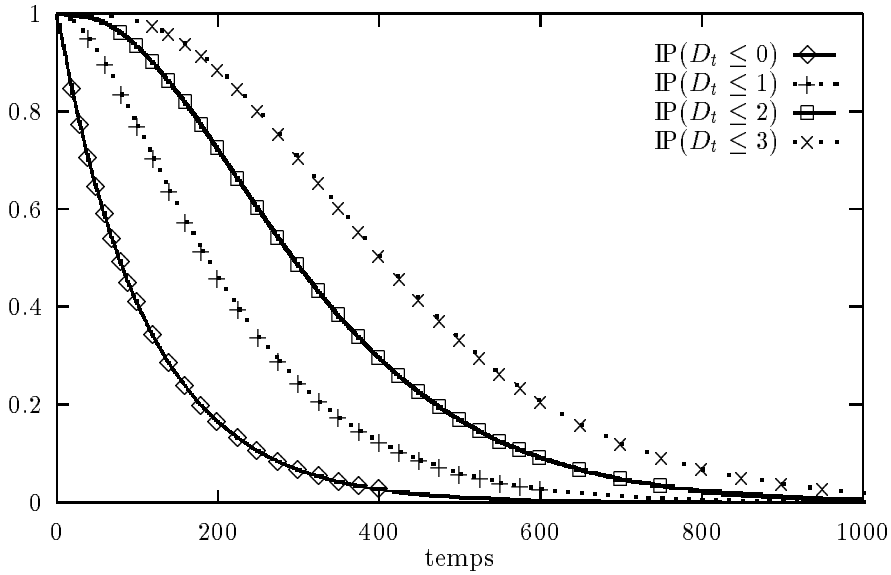


FIG. 3 - $(Op)_1$ *:La fonction* $(I\!P(D_t \leq k))_{t \geq 0}$ *pour différentes valeurs de* k

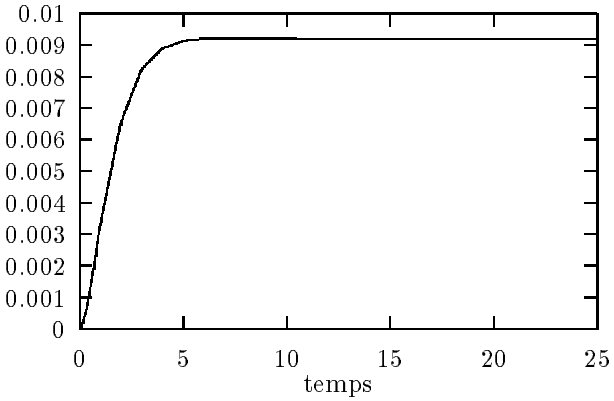


FIG. 4 - *Fonction intensité pour le modèle* $(Op)_1$

des caractéristiques 0-défaillance et 2-défaillances sont données dans la Figure 5 pour un taux résiduel λ_4 de 0.01 puis de 0.03.

Par ailleurs, nous pouvons facilement évaluer la perturbation sur la distribution du compteur $(D_t)_{t \geq 0}$ engendrée par le changement du premier composant exécuté après défaillance entre les modèles $(Op)_1$ et $(Op)_2$. Il est clair que la fonction $t \rightarrow \mathbb{P}(D_t \leq k)$ sera d'autant moins perturbée que le nombre de défaillances k considéré est faible. En particulier, pour $k = 0$, la fonction fiabilité est identique pour les deux modèles. L'influence de la redirection du contrôle d'exécution ne peut être évidente que si l'on considère l'occurrence de plusieurs incidents, c'est à dire que si les mécanismes de retour au bon fonctionnement sont activés sur $]0, t]$. Pour notre exemple, la différence entre les valeurs de $\mathbb{P}(D_t \leq k)$ pour $(Op)_1$ et $(Op)_2$ et $k = 1$ peut atteindre 10^{-2} sur approximativement l'intervalle $[50h, 250h]$, plus de 10^{-2} sur l'intervalle $[110h, 450h]$ avec $k = 2$ et à peu près 2.10^{-2} avec $k = 3$ sur l'intervalle $[200h, 450h]$.

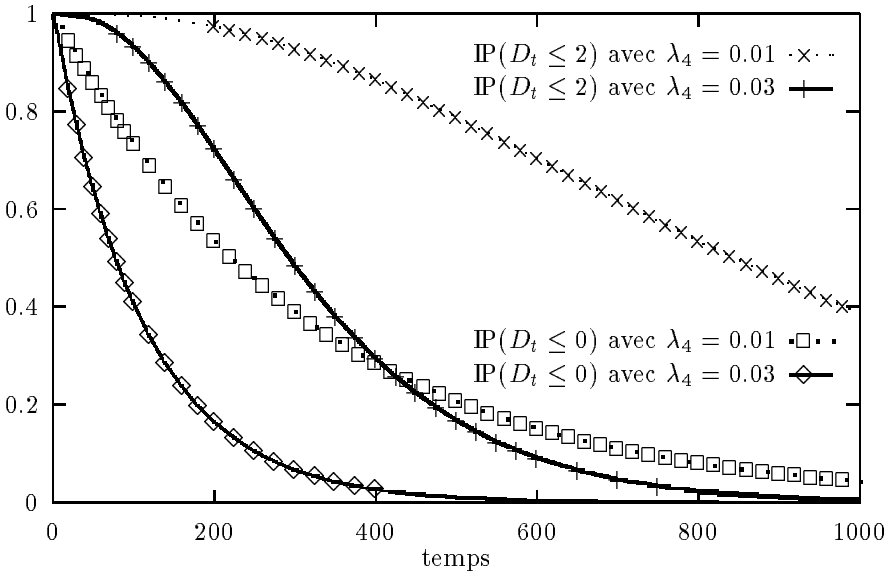


FIG. 5 - *La fonction* $(I\!P(D_t \leq k))_{t \geq 0}$ *pour les valeurs* 0.01 *et* 0.03 *de* λ_4 *dans le modèle* $(Op)_1$.

Par exemple si on reprend la question posée précédemment pour le second modèle, on obtient alors $t \leq 95h$. Ceci représente une différence d'environ 8% sur la borne supérieure de l'intervalle.

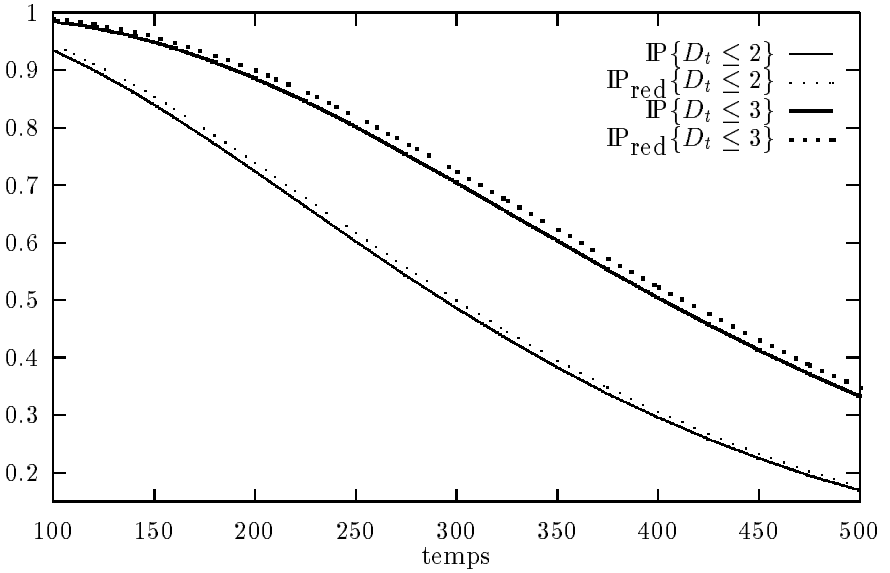


FIG. 6 - *La fonction* $(I\!P(D_t \leq k))_{t \geq 0}$ *pour* $k = 2, 3$ *pour les modèles* $(Op)_1$ *et* $(Op)_2$

4.2 Croissance de fiabilité

La caractéristique commune des modèles de fiabilité du logiciel de type "boîte-noire" est de simuler un phénomène de croissance de fiabilité. Ce dernier est généralement observé lorsque des corrections efficaces sont apportées au logiciel et en particulier aux différents composants. Dans une modélisation de type structurelle, l'introduction d'une telle éventualité n'a été réalisée que récemment par Laprie et al. [22]. Elle repose sur l'idée suivante : choisir un modèle de fiabilité de type "boîte noire" pour représenter la croissance de fiabilité d'un composant et établir des règles d'intégration dans le modèle structurel du logiciel. Le processus de défaillance du composant est un processus NHPP hyperexponentiel développé dans [15] et [13] et dont l'intensité de défaillance est donnée par

$$h(t) = \frac{d}{dt}\mathbb{E}[D_t] = \frac{\omega\lambda_{sup}e^{-\lambda_{sup}t} + (1-\omega)\lambda_{inf}e^{-\lambda_{inf}t}}{\omega e^{-\lambda_{sup}t} + (1-\omega)e^{-\lambda_{inf}t}},$$

où $0 \leq \lambda_{inf} \leq \lambda_{sup}$ et $0 \leq \omega \leq 1$.

L'intensité de défaillance décroît ainsi de $h(0) = \omega\lambda_{sup} + (1-\omega)\lambda_{inf}$ jusqu'à $h(\infty) = \lambda_{inf}$. Cette fonction peut être vue comme le taux de défaillance d'une distribution PH (ou distribution de Cox) à deux phases placées en parallèle (voir Figure 7 (b)). Dans le cadre des travaux poursuivis au LAAS [22],[13], le modèle structurel de croissance de fiablité est construit grâce à l'interprétation markovienne précédente du modèle NHPP hyperexponentiel.

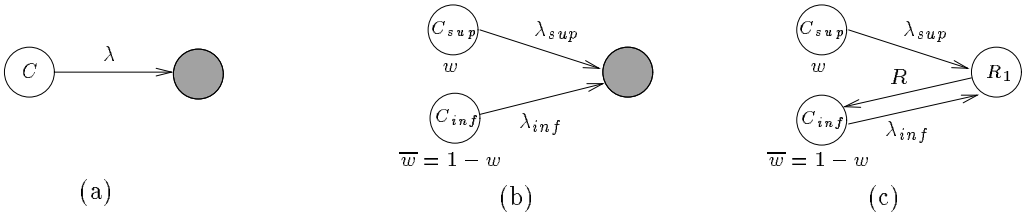


FIG. 7 - *Chaîne de Markov représentant un composant en situation de fiabilité stabilisé* (a), *en croissance de fiabilité* (b) *puis en croissance de disponibilité* (c).

Nous pouvons également simuler le phénomène de croissance de fiabilité de certains composants dans notre modèle, en reprenant le schéma de transformation de la Figure 7. Les graphes de transition des modèles $(Op)_1$ et $(Op)_2$ deviennent alors ceux représentés dans la Figure 8, lorsque la fiabilité du composant C_4 croît.

A partir de l'algorithme du Théorème 3.1 on constate qu'il y a décroissance stochastique de la distribution de D_t, c'est à dire, si $(D_t)_{\text{crois}}$ désigne le compteur de défaillances en situation de croissance de fiabilité de C_4,

$$D_t \overset{st}{\geq} (D_t)_{\text{crois}} \quad \text{i.e} \quad \mathbb{P}(D_t \leq k) \leq \mathbb{P}((D_t)_{\text{crois}} \leq k) \quad (\forall k \in \mathbb{N}).$$

Reprenons la question "jusqu'à quelle date t la probabilité d'observer au plus deux défaillances est plus grande que 0.95 ?" pour le modèle $(Op)_1$ et le modèle tenant compte de la croissance de fiabilité du composant C_4 avec les paramètres $w = 0.5$, $\lambda_{sup} = 0.0585$ et $\lambda_{inf} = 0.0015$. Ceci donne un rapport de croissance $\lambda_4/\lambda_{inf} = 20$ et une valeur initiale $h(0) = \lambda_4$. Nous avions déterminé $t \leq 88h$ pour $(Op)_1$ et maintenant nous obtenons $t \leq 97h$. Nous constatons que le gain sur la borne de l'intervalle est d'environ 10%.

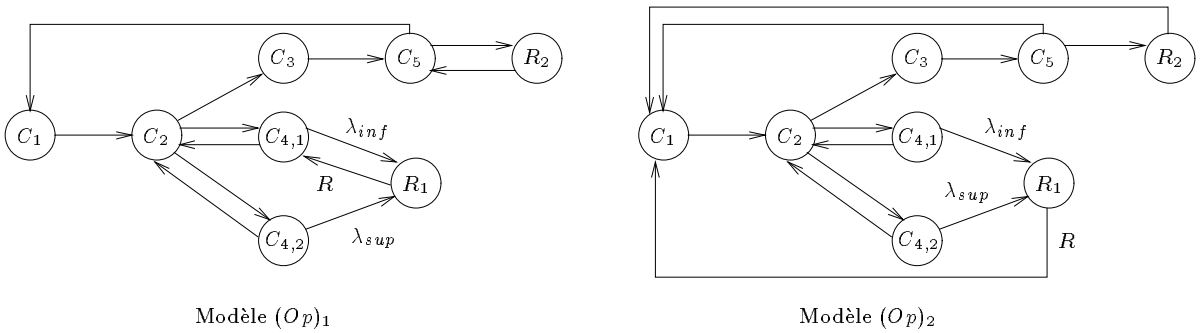


FIG. 8 - *Modèles opérationnels incluant une croissance de fiabilité du composant C_4.*

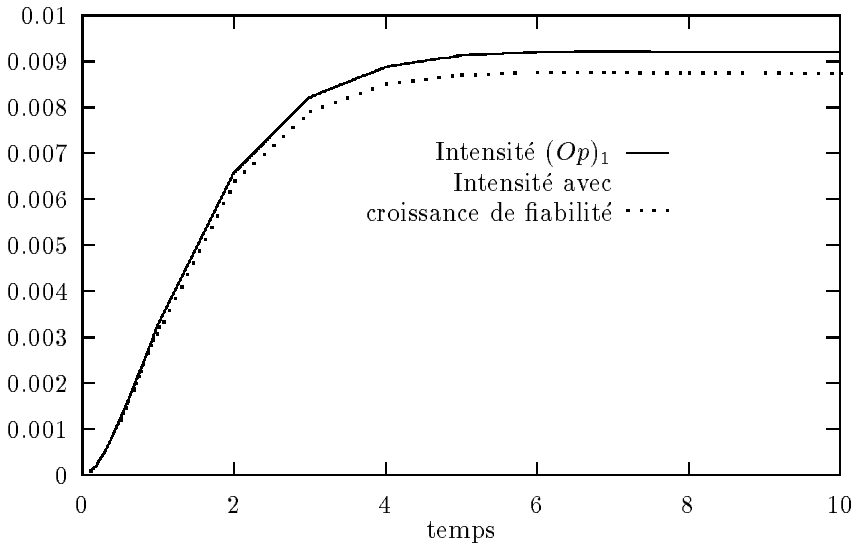


FIG. 9 - *Fonctions intensité pour les modèles $(Op)_1$ et $(Op)_1$ avec une croissance de fiabilité du composant C_4.*

4.3 Approximation poissonnienne

Nous avons vu que le temps T jusqu'à la première défaillance dans notre modèle était donné par

$$\mathbb{P}(T > t) = \mathbb{P}(D_t = 0) = \alpha e^{At} 1^{\mathrm{T}}.$$

Le processus (markovien) nominal X, de générateur Q a été supposé irréductible en Sous-section 3.1. Ceci implique que la matrice A est également un sous-générateur irréductible. Il est connu [32] qu'une distribution de type phase (PH) de sous-générateur irréductible, ici A, est asymptotiquement exponentielle de paramètre, l'unique valeur propre réelle λ, strictement négative et strictement plus grande que la partie réelle de toute autre valeur propre de la matrice A. Ce scalaire est parfois appelé la valeur propre de Perron-Frobenius de A. Ainsi, nous avons

$$\mathbb{P}(T > t) = e^{-\lambda t} + o(e^{-\lambda t}),$$

avec $o(e^{-\lambda t})/e^{-\lambda t} \to 0$ lorsque $t \to +\infty$. Le taux λ peut être interprété comme le taux de défaillance asymptotique du système lorsque $t \to +\infty$. En effet, nous avons par définition

$$\begin{aligned}\lambda(t) &= \mathbb{P}(\text{ une déf. survient sur } [t, t+dt[\mid \text{ aucune déf. sur } [0,t[\ ) \\ &= \sum_{i \in \mathcal{M}} \frac{\alpha e^{At}(i)}{\alpha e^{At} 1^{\mathrm{T}}} D^{\mathrm{T}}(i).\end{aligned}$$

On en déduit que pour $t \to +\infty$

$$\lambda(\infty) = \sum_{i \in \mathcal{M}} v(i) D^{\mathrm{T}}(i),$$

où v est la distribution dite quasi-stationnaire [37] associée au sous-générateur A, c'est à dire l'unique distribution de probabilité satisfaisant $vA = \lambda v$. On peut alors en déduire que $-\lambda = \lambda(\infty)$. On peut remarquer que pour tout $i \in \mathcal{M}$, $v(i)$ et $\pi(i)/\sum_{i \in \mathcal{M}} \pi(i)$ sont des quantités distinctes (π est la distribution stationnaire de X^*). Enfin si la distribution de T est asymptotiquement exponentielle, cela ne signifie pas que le processus de comptage des événements suit nécessairement une loi de Poisson de paramètre λ.

Nous avons déjà signalé que la fonction intensité de défaillance devient constante lorsque $t \to +\infty$ et que cette constante vaut

$$h(\infty) = \pi D^{\mathrm{T}}.$$

La quantité $\lambda_d^* = \pi D^{\mathrm{T}}$ a été appelée le taux fondamental de $(D_t)_{t \geq 0}$. En particulier, nous retrouvons pour le modèle de Littlewood ($R = S = 0$) le paramètre de la loi de Poisson approchant la distribution du compteur D_t. Rappelons que π est alors également la distribution stationnaire du modèle nominal X ($\pi Q = 0, \quad \pi 1^{\mathrm{T}} = 1$). Nous pouvons essayer de comparer le calcul de la distribution exacte de D_t pour le modèle $(Op)_1$ avec celle de Poisson de paramètre l'intensité asymptotique

$$\pi(4) D^{(p)}(4,6) + \pi(5) D^{(p)}(5,7) = \pi(4) 0.03 + \pi(5) 0.01.$$

On constate que l'approximation de type Poisson donne une très légère sous-estimation de la sûreté de fonctionnement du système à partir du modèle $(Op)_1$. On est donc assuré que les spécifications de fiabilité seront respectées à partir du calcul des mesures avec cette distribution de Poisson. Cependant, la même conclusion ne tient pas pour le modèle $(Op)_2$. En effet, le même type de comparaison montre que l'approximation poissonnienne sur-estime la sûreté de fonctionnement du système, puis la sous-estime. Nous ne pouvons ainsi en déduire aucune relation d'ordre a priori entre les indicateurs de fiabilité obtenus à partir du processus de Poisson et ceux issus du modèle exact initial.

Cette approximation de type Poisson est justifiée dans le contexte des articles [27] et [28] sous la seule condition que $(\mu(i,j),\ \lambda_j)_{i,j \in \mathcal{M}}$ tendent vers 0. Dans ce cas, on peut s'attendre effectivement à ce que la date d'occurrence d'une défaillance soit stochastiquement croissante et, par conséquent, à ce que le processus d'exécution soit proche de son régime stationnaire lorsque cet incident surviendra. Cependant, l'énoncé du théorème peut paraître surprenant dans le modèle de Lillewood car le processus de défaillance n'a aucune influence sur le processus identité ou d'exécution ($X^* = X$). Prenons le cas particulier où le système est constitué de deux composants avec occurrence uniquement de défaillances internes secondaires (cf Exemple 2). Nous avons alors pour l'intensité

$$h(t) = \alpha e^{Q^* t} D^{(s)} 1^{\mathrm{T}} = \alpha e^{Qt} D^{(s)} 1^{\mathrm{T}}.$$

Elle peut être réécrite sous la forme

$$h(t) = \pi D^{(s)} 1^{\mathrm{T}} + [\alpha(1)\pi(2) - \alpha(2)\pi(1)] e^{(Q(1,1)+Q(2,2))t} (\mu_1 + \mu_2).$$

Il est clair que s'il y a convergence vers une distribution de Poisson de paramètre $\lambda^* = \pi D^{(s)} 1^{\mathrm{T}}$, elle ne peut avoir lieu sous la seule condition de décroissance vers 0 de l'amplitude des paramètres μ_1, μ_2. Il

semble que le résultat ait été donné sous l'hypothèse d'un processus d'exécution en régime stationnaire. Ceci correspond alors à étudier la version stationnaire du processus ponctuel des défaillances (voir [24] pour une étude dans le contexte de fiabilité ou [33] dans le cadre général des processus d'arrivée d'une file d'attente). Le même genre de remarque vaut pour la fonction fiabilité $\mathbb{P}(T > t)$ (voir [24]).

Enfin, lorsque la période d'observation t devient très grande, les formules (4) et (5) nous fournissent des représentations asymptotiques des valeurs moyennes. Pour un processus markovien, on sait que la probabilité asymptotique $\pi(i)$ d'occuper l'état i représente la proportion (avec t grand) de temps passé dans i sur une unité de temps, c'est à dire $\pi(i) = \lim_{t\to\infty}(\mathbb{E}[S_i(t)]/t)$. On déduit de l'équation (4) que

$$\lim_{t\to\infty}\frac{\mathbb{E}[D_t]}{t} = \sum_{i\in\mathcal{M}}\lim_{t\to\infty}\frac{\mathbb{E}[S_i(t)]}{t}D^{\mathrm{T}}(i) = \pi D^{\mathrm{T}} = \lambda_d^*.$$

La formulation (5) de la fonction $\mathbb{E}[D_t]$ donne l'équation de son asymptote linéaire :

$$\mathbb{E}[D_t] = \lambda_d^* t + [(\alpha,0) - \pi](1^{\mathrm{T}}\pi - Q^*)^{-1}D^{\mathrm{T}} + o(1).$$

Nous constatons que si on considère la version stationnaire du processus ponctuel des défaillances (i.e. la distribution initiale $(\alpha,0)$ de X^* est remplacée par sa distribution stationnaire π), on a

$$\mathbb{E}[D_t] = \lambda_d^* t \qquad \forall t \in \mathbb{R}^+.$$

Nous avons alors comme intensité de défaillance

$$h(t) = \frac{d\mathbb{E}[D_t]}{dt} = \lambda_d^*,$$

c'est à dire l'intensité asymptotique du modèle.

Conclusion

Ce papier propose l'étude transitoire du processus de comptage des défaillances d'un modèle structurel markovien d'un logiciel. Après avoir examinées les conséquences d'une approche markovienne pour représenter le comportement d'un système logiciel, nous obtenons la fonction de répartition et l'espérance du nombre de défaillances observées sur un intervalle de mission ainsi que les expressions analytiques d'autres grandeurs classiques en fiabilité. Les délais de retour à un état opérationnel après une interruption de service peuvent dépendre de l'identité du composant défaillant et sont supposés distribués selon une loi de type phase. Il est important de noter que les métriques de sûreté de fonctionnement issues de l'application de la Section 3 sont plus informatives que celles données habituellement car elles donnent une image du comportement du système sur un intervalle de mission. De plus, les temps de calcul requis avec l'algorithmique proposée sont faibles car ils n'invoquent que des produits vecteurs-matrices de petite dimension et de faible densité en coefficients non nuls. Un certain nombre de propriétés asymptotiques du processus de comptage des d'efaillances, ainsi que l'introduction du phénomène de croissance de fiabilité de certains composants dans notre modèle sont également discutés. L'analyse du modèle permet également de valider "expérimentalement" le comportement poissonnien justifié par Littlewood dans une situation où le système est déjà sûr de fonctionnement. Enfin, l'apport d'une telle approche structurelle est illustrée sur quelques exemples.

La limite majeure à la validation des modèles de fiabilité du logiciel a longtemps été le manque de données publiées. Il semble que ceci ait été surmonté dans le contexte d'une modélisation boîte-noire [2],[4]. Cependant, à notre connaissance, il n'existe aucune donnée publique qui puisse supporter la validation générale du modèle structurel proposé. Au vu de la richesse des informations à collecter, il est clair que le nombre final de composants devra rester le plus petit possible. Ceci signifie que l'on devra rechercher un compromis entre la puissance de description du modèle et le problème d'estimation de ses paramètres.

Références

[1] N. Ashrafi and F. Zahedi. Software reliability allocation based on structure, utility, price, and cost. *IEEE Trans. Software Eng.*, 17(4):345–356, 1991.

[2] V.R. Basili and D.M. Weiss. A methodology for collecting valid software engineering data. *IEEE Trans. Software Eng.*, 10:728–738, 1984.

[3] R.C. Cheung. A user-oriented software reliability model. *IEEE Trans. Software Eng.*, 6(2):118–125, 1980.

[4] P. Comer. Software data collection and the software data library. In *6th EUREDATA Conference*, pages 824–839, 1989.

[5] E. de Souza e Silva and H.R. Gail. Calculating availability and performability measures of repairable computer systems using randomization. *Journal of the ACM*, 36(1):171–193, 1989.

[6] E. de Souza e Silva and H.R. Gail. Performability analysis of computer systems: from model specification to solution. *Perf. Evaluation*, 14:157–196, 1992.

[7] M.J. Faddy. Examples of fitting structured phase-type distribution. *Applied Stochastic Models and Data Analysis*, 10:247–255, 1994.

[8] O. Gaudoin. *Outils Statistiques pour l'Evaluation de la Fiabilité des Logiciels.* PhD thesis, Université Joseph Fourier - Grenoble I, 1990.

[9] D. Gross and D.R. Miller. The randomization technique as a modeling tool and solution procedure for transient Markov processes. *Operations Research*, 32(2):343–361, 1984.

[10] H. Heffes and D.M. Lucantoni. A Markov Modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE Trans. Selected areas in Comm.*, 4(6):856–868, 1986.

[11] A. Iannino et al. *Software Reliability: Measurement, Prediction, Application.* McGraw-Hill International Editions, Computer Science Series, 1987.

[12] Raj Jain. *The art of computer systems performance analysis.* John Wiley & Sons, Inc., 1991.

[13] M. Kaâniche. *Modèle Hyperexponentiel en Temps Continu et en Temps Discret pour l'Evaluation de la Croissance de la Sûreté de Fonctionnement.* PhD thesis, LAAS, Toulouse, 1992.

[14] M. Kaâniche and K. Kanoun. The discrete time hyperexponential model for software reliability growth evaluation. In *Int. Symp. on Software Reliability (ISSRE)*, Oct. 1992.

[15] K. Kanoun. Croissance de la sûreté de fonctionnement des logiciels caractérisation-modélisation-evaluation. Technical Report 89.320, Thèse de doctorat d'état (LAAS), Toulouse, 1989.

[16] K. Kanoun et al. Experience in software reliability : from data collection to quantitative evaluation. Technical Report 93-114, LAAS, Mars 1993.

[17] K. Kim and K.S. Park. Phased-mission system under Markov environment. *IEEE Trans. Reliability*, 43(2):301–309, 1994.

[18] P. Kubat. Assessing reliability of modular software. *Operations Research Letters*, 8:35–41, 1989.

[19] Laprie et al. Modèles structurels et croissance de fiabilité. Technical Report 95.279, LAAS, Juin 1995.

[20] J.C. Laprie. Dependability evaluation of software systems in operation. *IEEE Trans. Software Eng.*, SE-16(6):701–714, 1984.

[21] J.C. Laprie. *Dependability: Basic Concepts and Terminology.* Springer-Verlag, Vienna, 1992.

[22] J.C. Laprie et al. The KAT (knowledge-action-transformation) approach to the modeling and evaluation of reliability and availability growth. *IEEE Trans. Software Eng.*, 17(4):370–382, 1991.

[23] J.C. Laprie and K. Kanoun. X-ware reliability and availability modeling. *IEEE Trans. Software Eng.*, 18(2):130–147, 1992.

[24] J. Ledoux. *Modèles markoviens: sur la caracterisation de l'agrégation faible et sur les modèles structurels pour la sûreté de fonctionnement du logiciel.* PhD thesis, No 1066, Université de Rennes I, 1993.

[25] J. Ledoux. Principaux modèles d'évaluation de la fiabilité du logiciel et techniques de validation de systèmes de prédiction: étude bibliographique. Technical Report 667, IRISA, Juillet 1992.

[26] J. Ledoux and G. Rubino. A counting model for software reliability analysis. Technical Report 2060, INRIA, Campus de Beaulieu, 35042 Rennes Cedex, France, 1993.

[27] B. Littlewood. A reliability model for systems with Markov structure. *Appl. Statist.*, 24(2):172–177, 1975.

[28] B. Littlewood. Software reliability model for modular program structure. *IEEE Trans. Reliability*, R-28(3):241–246, 1979.

[29] D.M. Lucantoni. New results on the single server queue with a batch markovian arrival process. *Commun. Statist. – Stochastic Models*, 7(1):1–46, 1991.

[30] Malhotra and Reibman. Selecting and implementing phase approximations for semi-Markov models. *Comm. Statist.- Stochastic Models*, 1993.

[31] Y. Masuda et al. A statistical approach for determining release time of software system with modular structure. *IEEE Trans. Reliability*, 38(3):365–372, 1988.

[32] M.F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach.* The John Hopkins University Press, 1981.

[33] M.F. Neuts. *Structured Stochastic Matrices of M/G/1 Type and Their Applications.* Marcel Dekker Inc., New-York and Basel, 1989.

[34] D. Noyes. Systèmes à missions phasées et modèles stochastiques. *RAIRO APII*, 26(4):277–293, 1992.

[35] A. Pagès and M. Gondran. *Fiabilité des systèmes.* Eyrolles, 1980.

[36] G. Rubino and B. Sericola. Interval availability distribution computation. In *FTCS*-23, pages 48–55, Juin 1993.

[37] E. Seneta. *Non-negative matrices and Markov chains.* Springer-Verlag, 1981.

[38] K. Siegrist. Reliability of systems with Markov transfer of control. *IEEE Trans. Software Eng.*, 14(7):1049–1053, 1988.

[39] K. Siegrist. Reliability of systems with Markov transfer of control, II. *IEEE Trans. Software Eng.*, 14(10):1478–1480, 1988.

[40] M. Smotherman and K. Zemoudeh. A non-homogeneous Markov model for phased-mission reliability analysis. *IEEE Trans. Reliability*, 38(5):585–590, 1989.

[41] D.L. Snyder. *Random Point Processes*. Wiley-Interscience Publication, 1975.

[42] M. Xie. *Software Reliability Modelling*. World Scientific Publishing, UK office: 73 Lynton Mead, Totteridge, London N20 8DH, 1991.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399